

ISO TC184/SC4/WG10 Nxxx

Date: April 8, 1997

Supersedes SC4/_____ N _____

PRODUCT DATA REPRESENTATION AND EXCHANGE

Part: **Title:** Part 203 ARM EXPRESS

Purpose of this document as it relates to the target document is:

- ☒ Primary Content
☐ Issue Discussion
☐ Alternate Proposal
☐ Partial Content

Current Status: Complete

ABSTRACT:

A draft rendition of the Part 203 ARM in EXPRESS.

KEYWORDS:

AP Interoperability,
Implementable ARM,
ARM module

Document Status/Dates (dd/mm/yy)

Part Documents	Other SC4 Documents
_____	Project Leader _____
_____	Working Group Convenor _____
_____	Qualification _____
_____	Integration _____
_____	Editing _____

Owner/Editor: P. R. Kennicott

Address: P.O. Box 633
Sandia Park, NM 87047

Alternate:

Address:

Telephone/FAX: 505 281 9581

E-mail:

Telephone/FAX:

E-mail:

Comments to Reader

This document completes the author's action with respect to the definition of modules for constructing ARMs with the corresponding AIM data being defined. The remaining work is to complete the mapping and replace the DATA_TYPE domains with appropriate references.

:

Contents	Page
1 Introduction	1
2 Information Requirements	1
2.1 Part Identification and Source Control	1
2.1.1 Part	4
2.1.2 Part_type_enum	4
2.1.3 Alternate_part	5
2.1.4 Part_version	5
2.1.5 Make_or_buy	7
2.1.6 Status_of_release	7
2.1.7 Substitute_part	7
2.1.8 Supplier	8
2.1.9 Supplied_part_version	8
2.2 Design Information	9
2.2.1 Additional_design_information	9
2.2.2 Design_specification	11
2.2.3 Material_specification	11
2.2.4 Process_specification	12
2.2.5 Specification	12
2.2.6 Surface_finish_specification	13
2.2.7 Usage_constraint	13
2.3 Shape	13
2.3.1 Geometric_model_representation	14
2.3.2 Component_assembly_position	14
2.3.3 Advanced_B_rep	16
2.3.4 Faceted_B_rep	16
2.3.5 Manifold_surface_with_topology	16
2.3.6 Non_topological_surface_and_wireframe	17
2.3.7 Wireframe_with_topology	17
2.4 Effectivity and End-Item Identification	17
2.4.1 Planned_effectivity	17
2.4.2 Planned_date_effectivity	19
2.4.3 Planned_lot_effectivity	19
2.4.4 Planned_sequence_effectivity	20
2.4.5 Product_configuration	21
2.4.6 Product_model	21
2.5 Design Activity Control	22
2.5.1 Change_order	22
2.5.2 Change_request	22
2.5.3 Start_order	24
2.5.4 Start_request	24
2.5.5 Work_order	25
2.5.6 Work_request	25
2.5.7 Request_status	26

2.6	Bill of Material	27
2.6.1	Engineering_assembly	27
2.6.2	Design_discipline_product_definition	29
2.6.3	Engineering_make_from	30
2.6.4	Engineering_next_higher_assembly	30
2.6.5	Engineering_promissory_usage	31
2.6.6	Shape	32
2.6.7	Shape_aspect	32
2.7	Authorization	33
2.7.1	Approval	33
2.7.2	Person_organization	33
3	The 203 ARM Objects	35

Figures

1	Controlled_configuration_ARM schema	2
2	Part Identification and Source Control	3
3	Design Information	10
4	Shape	15
5	Effectivity and End-Item Identification	18
6	Design Activity Control	23
7	Bill of Material	28
8	Authorization	34

:

1 Introduction

This document specifies the information requirements for the exchange of configuration- controlled 3D designs of mechanical parts and assemblies in terms of an EXPRESS representation of the ARM.

The information requirements are specified as a set of units of functionality and application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTES

1 – A graphical representation of the information requirements is given in figures accompanying the document.

2 – The information requirements correspond to those of the activities identified as being in the scope of this application protocol in annex F of ISO 10303, Part 203.

3 – The mapping table is specified in Clause 5.1 of Part 203 which shows how the information requirements are met using the integrated resources of this International Standard. The use of the integrated resources introduces additional requirements which are common to application protocols.

This schema was derived (more or less) automatically from the IS version of Part 203. The overall schema is shown in Figure 1

EXPRESS specification:

```
*)
SCHEMA Controlled_configuration_arm;
(*
```

2 Information Requirements

This clause presents the various Units of Functionality (UoFs). Note that they do not necessarily correspond to those in Part 203. The UoFs in this document were composed using tools which determine the connectivity of EXPRESS objects.

2.1 Part Identification and Source Control

The Part Identification and Source Control UoF contains the structure through which a part, unique versions of a part, and different design discipline views of a part can be defined. It also contains the information about an organization that is qualified to produce a particular part.

The graphic representation of the Part Identification and Source Control UoF is shown in Figure 2.

:

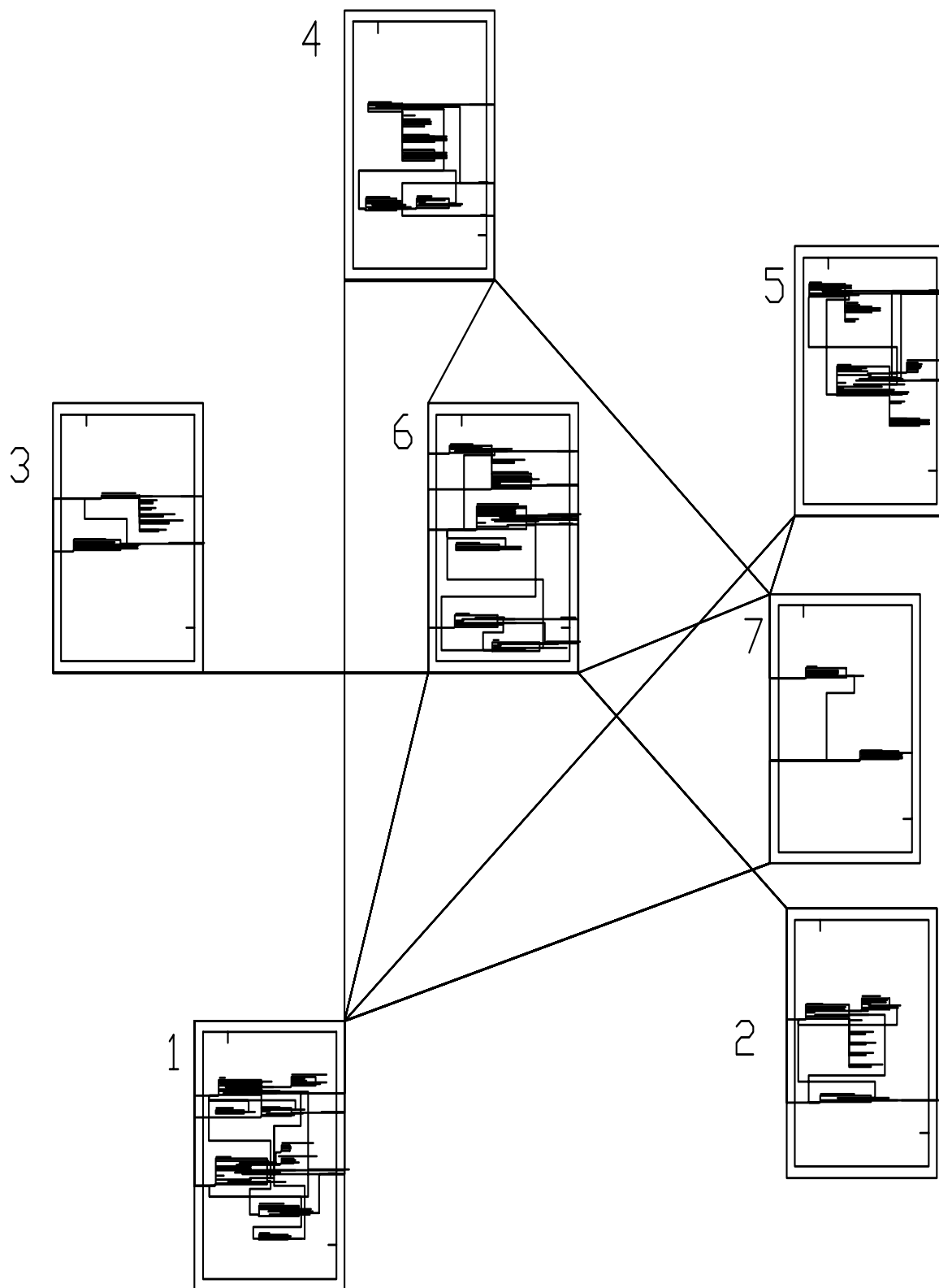


Figure 1 – Controlled_configuration_ARM schema

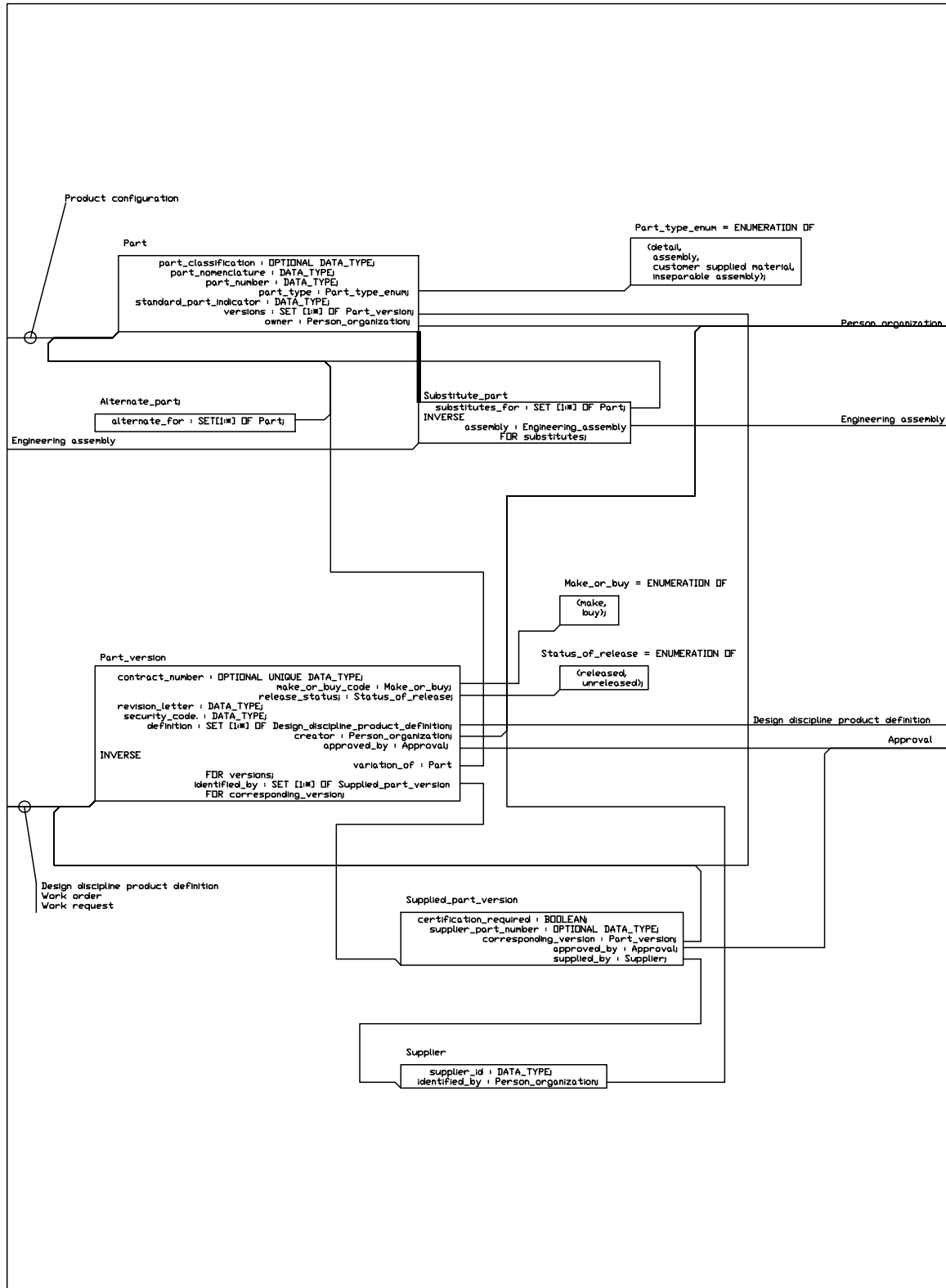


Figure 2 – Part Identification and Source Control

:

2.1.1 Part

A Part is an item that is intended to be produced or employed in a production process.

EXPRESS specification:

```
*)
ENTITY Part
    part_classification : OPTIONAL DATA_TYPE;
    part_nomenclature : DATA_TYPE;
    part_number : DATA_TYPE;
    part_type : Part_type_enum;
    standard_part_indicator : DATA_TYPE;
    versions : SET [1:*) OF Part_version;
    owner : Person_organization;
END_ENTITY;
(*
```

Attribute definitions:

part_classification: a family of related parts that have common processes applied to them during manufacturing. The **part_classification** need not be specified for a particular Part.

EXAMPLES

1 – Machined, punched, or turned parts may be categorized into separate part classes.

part_nomenclature: a name by which the part is commonly known within an organization.

NOTE – The name can also have meaning outside of any single organization. For example, a single nation or industry can have naming conventions for different kinds of common parts.

part_number: the unique identification of a part for a particular organization.

part_type: one of a set of kinds of parts.

standard_part_indicator: a specification of whether or not a part has a design specified externally to the bill of material in which an assembly is being defined.

NOTE – A part may be a standard part for a company, industry, or other organization.

versions: the set of **Part_version** objects which define variations of this **Part**.

owner: the **Person_organization** which is the owner of this **Part**.

2.1.2 Part_type_enum

This enumeration type specifies the values for the kinds of parts identified for Part 203

EXPRESS specification:

:

```
*)
TYPE Part_type_enum = ENUMERATION OF
  (detail,
   assembly,
   customer supplied material,
   inseparable assembly};
END_TYPE;
(*
```

Enumerated item definitions:

detail: a part that exists at the lowest level of the bill of material.

assembly: a part that consists of a collection of other parts that are put together in order to satisfy a particular function.

customer supplied material: a part that has been furnished to the design from a customer source.

inseparable assembly: an assembly that, after assembly, cannot be dis-assembled without causing physical harm to at least one of the components of the assembly.

2.1.3 Alternate_part

An **Alternate_part** is a part that is interchangeable with another part with respect to form, fit, and function.

NOTE – The usages of an alternate part are not of interest to an organization that owns it and therefore are not tracked by that organization.

EXAMPLES

2 – A design requires a sheet metal screw of a certain size. Sheet metal screws are made by various organizations and are equivalent with respect to form, fit, and function for that size. The various screws have different part numbers assigned to them by the manufacturer. The design organization does not track which screw is used in the actual assembly. These screws are alternate parts.

EXPRESS specification:

```
*)
ENTITY Alternate_part;
  alternate_for : SET[1:~] OF Part;
END_ENTITY;
(*
```

Attribute definitions:

alternate_for: The set of **Parts** for which this **Alternate_part** is an alternate.

2.1.4 Part_version

A **Part_version** is the identification of the representation a part after its design has undergone a formal release or change. Only changes that are formally tracked by the organization that

:

is responsible for the definition of the part shall result in a new instance of this application object. Iterations on designs that are not formally tracked by the organization responsible for the definition of the part shall not result in a new instance of this application object, but shall be tracked by the **Design_discipline_product_definition** application object.

EXPRESS specification:

```
*)
ENTITY Part_version
    contract_number : OPTIONAL UNIQUE DATA_TYPE;
    make_or_buy_code : Make_or_buy;
    release_status; : Status_of_release;
    revision_letter : DATA_TYPE;
    security_code. : DATA_TYPE;
        definition : SET [1:~] OF Design_discipline_product_definition;
        creator : Person_organization;
        approved_by : Approval;
INVERSE
    variation_of : Part
                FOR versions;
    identified_by : SET [1:~] OF Supplied_part_version
                FOR corresponding_version;
END_ENTITY;
(*
```

Attribute definitions:

contract_number: the business contract under which the part was designed. The **contract_number** need not be specified for a particular **Part_version**. If the design was developed under a contract, this attribute shall specify the unique identification of that contract.

make_or_buy_code: the design organization's plan for obtaining the part.

release_status: the status of the version of a part with respect to the dissemination of design information.

revision_letter: the unique identification of a particular version of a part.

security_code: the security classification of a particular version of a part.

definition: the **Design_discipline_product_definition** which defines this **Part_version**.

creator: the **Person_organization** which created this **Part_version**.

approved_by: the **Approval** which approves this **Part_version**.

variation_of: the **Part** of which this **Part_version** is a variation.

identified_by: the set of **Supplied_part_version** objects which correspond to this **Part_version**.

2.1.5 Make_or_buy

The **Make_or_buy** enumeration type defines the allowable values for the **make_or_buy** attribute of **Part_version**.

EXPRESS specification:

```
*)
TYPE Make_or_buy = ENUMERATION OF
    (make,
     buy};
END_TYPE;
(*
```

Enumerated item definitions:

make: the design organization has planned to make the part internally.

buy: the design organization plans for the part to be purchased from a vendor.

2.1.6 Status_of_release

The **Status_of_release** enumeration type defines the value for the **release_status** attribute of **Part_version**.

EXPRESS specification:

```
*)
TYPE Status_of_release = ENUMERATION OF
    (released,
     unreleased);
END_TYPE;
(*
```

Enumerated item definitions:

released: those versions that have been reviewed and approved for further use.

unreleased: those versions that have been neither reviewed nor approved for further use.

2.1.7 Substitute_part

A **Substitute_part** is a component within an assembly whose form, fit, and function might be different from the component for which it is a replacement, but that can fulfill the requirements of another part within the context of the assembly.

NOTES

1 – A substitute part is not required to have the same shape or quantity as the original part.

2 – This application object also may be used to eliminate version roll-up, i.e., the re-identification of all higher level assemblies when a new version of a lower-level component is created.

:

3 – This application object allows one-way substitution only. Within a given context, if A is a substitute for B, then B is not to be assumed a substitute for A, unless explicitly stated so in another instance of this entity. If A is a substitute for B within context C, then either A or B may be used within an eventually built unit for C, but not both.

EXAMPLES

3 – A gauge of 120 psi \pm 12 psi may be substituted by a gauge of 125 psi \pm 6.25 psi, provided the increments are equal and the input and output sizes, as well as the commodity it measures, are the same.

EXPRESS specification:

```
*)
ENTITY Substitute_part
SUBTYPE OF (Part);
    substitutes_for : SET [1:*) OF Part;
INVERSE
    assembly : Engineering_assembly
        FOR substitutes;
END_ENTITY;
(*
```

Attribute definitions:

substitutes_for: the set of **Parts** for which this **Substitute_part** is a possible replacement.

assembly: The **Engineering_assembly** in which this **Substitute_part** may replace a **Part**.

2.1.8 Supplier

A **Supplier** is the organization that produces the part or the design of the part.

EXPRESS specification:

```
*)
ENTITY Supplier
    supplier_id : DATA_TYPE;
    identified_by : Person_organization;
END_ENTITY;
(*
```

Attribute definitions:

supplier_id: the unique identification of the producing organization. This identification may be an organization-unique code, an industry- unique code or internationally recognized code.

identified_by: the **Person_organization** which identifies this **Supplier**.

2.1.9 Supplied_part_version

A **Supplied_part_version** is a part that is defined by its part number and its supplier.

EXPRESS specification:

:

```
*)
ENTITY Supplied_part_version
  certification_required : BOOLEAN;
  supplier_part_number : OPTIONAL DATA_TYPE;
  corresponding_version : Part_version;
  approved_by : Approval;
  supplied_by : Supplier;
END_ENTITY;
(*
```

Attribute definitions:

certification_required: the specification that a supplier must be approved prior to the procurement of a part from that supplier.

supplier_part_number: the part number used within the producing organization. The **supplier_part_number** need not be specified for a particular **Supplied_part_version**.

EXAMPLES

4 – Company A uses part 1234567-1 as part of an assembly. This part is actually purchased from company B, but additional testing is performed on each item received to assure it meets the requirements of the assembly. Company B identifies this part as AABBCCD-E. Company A identifies Company B as supplier number 52088. Within Company A, the part is identified as a Part with a part number of 1234567-1. The supplier is company B. The supplier code is 52088, and the supplier part number is AABBCCD-E.

corresponding_version: the **Part_version** corresponding to this **Supplied_part_version**.

approved_by: the **Approval** which approves this **Supplied_part_version**.

supplied_by: the **Supplier** for this **Supplied_part_version**.

2.2 Design Information

The Design Information UoF contains a collection of information about a part, separate from the part identification and shape information, that defines specific qualities the completed part shall possess. This information identifies specifications related to the design and manufacture of the part and constraints on the application of those specifications.

The graphic representation of the Design Information UoF is shown in Figure 3.

2.2.1 Additional_design_information

An **Additional_design_information** is a collection of specifications that are associated with the design of a part.

EXAMPLES

:

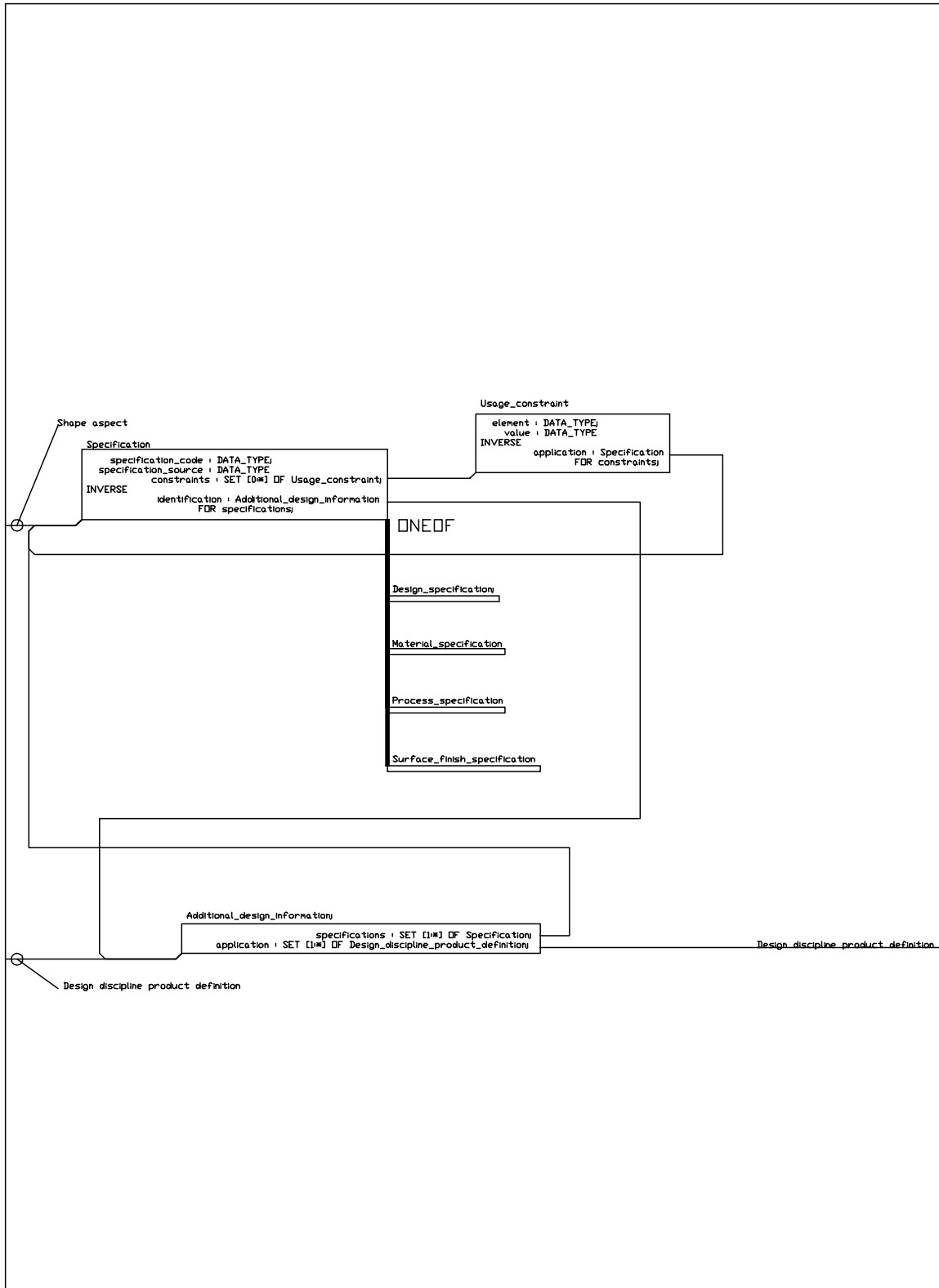


Figure 3 – Design Information

:

5 – The concepts of transmissivity, absorptivity, and reflectivity are additional design information in the form of product requirements which can be made know to downstream processes such as manufacturing and testing by a specification reference with a usage constraint related to the part.

EXPRESS specification:

```
*)
ENTITY Additional_design_information;
    specifications : SET [1:∗] OF Specification;
    application : SET [1:∗] OF Design_discipline_product_definition;
END_ENTITY;
(*
```

Attribute definitions:

specifications: the set of Specifications making up this **Additional_design_information**.

application: the set of **Design_discipline_product_definitions** to which this **Additional_design_information** applies.

2.2.2 Design_specification

A **Design_specification** is a type of **Specification** (see 2.2.5) that sets the design requirements of parts. These requirements are not determined by other design or construction features or referenced data.

EXAMPLES

6 – The requirements set forth in a **Design_specification** might be weight limits, dimensional limits, reflectivity, color, transparency, feel, and appearance.

EXPRESS specification:

```
*)
ENTITY Design_specification
SUBTYPE OF (Specification);
END_ENTITY;
(*
```

2.2.3 Material_specification

A **Material_specification** is a type of **Specification** (see 2.2.5) that specifies properties that are applicable to raw material, mixtures, or semi-fabricated material that are used in the fabrication of a product.

EXPRESS specification:

```
*)
ENTITY Material_specification
SUBTYPE OF (Specification);
END_ENTITY;
(*
```

:

2.2.4 Process_specification

A **Process_specification** is a type of **Specification** (see 2.2.5) that is the identification of a service to be performed on a product or material.

EXAMPLES

7 – Typical processes are heat treatment, welding, plating, packing, and marking.

EXPRESS specification:

```
*)  
ENTITY Process_specification  
SUBTYPE OF (Specification);  
END_ENTITY;  
(*
```

2.2.5 Specification

A **Specification** is a document that contains definitions, processes, or rules related to a unique quality that an in-process or finished part shall possess.

EXAMPLES

8 – Material, surface finish, and process specification are all examples of things that can be described implicitly by reference to other documents, rather than explicitly each time they are used.

EXPRESS specification:

```
*)  
ENTITY Specification  
SUPERTYPE OF ONEOF  
  (Design_specification,  
   Material_specification,  
   Process_specification,  
   Surface_finish_specification);  
  specification_code : DATA_TYPE;  
  specification_source : DATA_TYPE  
    constraints : SET [0:~] OF Usage_constraint;  
INVERSE  
  identification : Additional_design_information  
    FOR specifications;  
END_ENTITY;  
(*
```

Attribute definitions:

specification_code: a unique identification of a **Specification**. The code is unique within the source organization.

specification_source: the organization that is responsible for the **Specification**.

identification: The **Additional_design_information** which identifies this **Specification**.

constraints: The set of **Usage_constraints** which constrain this **Specification**.

2.2.6 Surface_finish_specification

A **Surface_finish_specification** is a type of **Specification** (see 2.2.5) that specifies properties that are applicable to surface textures or protective coatings for either an in-process or completed part.

EXPRESS specification:

```
*)
ENTITY Surface_finish_specification
SUBTYPE OF (Specification);
END_ENTITY;
(*
```

2.2.7 Usage_constraint

A **Usage_constraint** is a restriction of the application of a **Specification** (see 2.2.5). It associates a particular topic of a specification with specific information or text to be used in its application for a particular instance.

EXPRESS specification:

```
*)
ENTITY Usage_constraint
  element : DATA_TYPE;
  value : DATA_TYPE
INVERSE
  application : Specification
              FOR constraints;
END_ENTITY;
(*
```

Attribute definitions:

element: the particular section or topic in a **specification** that is being constrained or clarified.

value: limiting or particular data to be assigned to a section or topic in an element of a **Usage_constraint**.

EXAMPLES

9 – A part may require painting per a company written surface finish specification. The population would be: **Specification_code** = ABCD-1; **Specification_source** = XYZ Company; the paired **Usage_constraint** would be: **element** = colour; **value** = green; **element** = coats; **value** = 3.

application: the **Specification** which this **Usage_constraint** constrains.

2.3 Shape

The Shape UoF contains the geometric and topological definition of a part. This UoF permits different types of geometric representation for each portion of the shape of the part. These portions, when taken together, form the entire shape of the part.

:

The graphic representation of the Shape UoF is shown in Figure 4.

2.3.1 Geometric_model_representation

A **Geometric_model_representation** is the definition of the shape or a portion of the shape of a part.

EXPRESS specification:

```
*)
ENTITY Geometric_model_representation
SUPERTYPE OF (Advanced_B_Rep,
               Faceted_B_rep,
               Non_topological_surface_and_wireframe,
               Manifold_surface_with_topology,
               Wireframe_with_topology);
    shape_aspect_representation : Shape_aspect;
END_ENTITY;
(*
```

Attribute definitions:

shape_aspect_representation: the **Shape_aspect** represented by this **Geometric_model_representation**.

2.3.2 Component_assembly_position

A **Component_assembly_position** is the identification of the position of the component in the assembly.

EXPRESS specification:

```
*)
ENTITY Component_assembly_position;
    transformation : DATA_TYPE;
    higher_assembly : Engineering_next_higher_assembly;
    parent_representation : Geometric_model_representation;
    parent_assembly : Geometric_model_representation;
END_ENTITY;
(*
```

Attribute definitions:

transformation: the translation and orientation of the component in the geometric space of the assembly.

higher_assembly: the **Engineering_next_higher_assembly** at which this **Component_assembly_position** is located.

parent_representation: the **Geometric_model_representation** in which this **Component_assembly_position** is a component.

parent_assembly: the **Geometric_model_representation** containing this **Component_assembly_position** as an assembly.

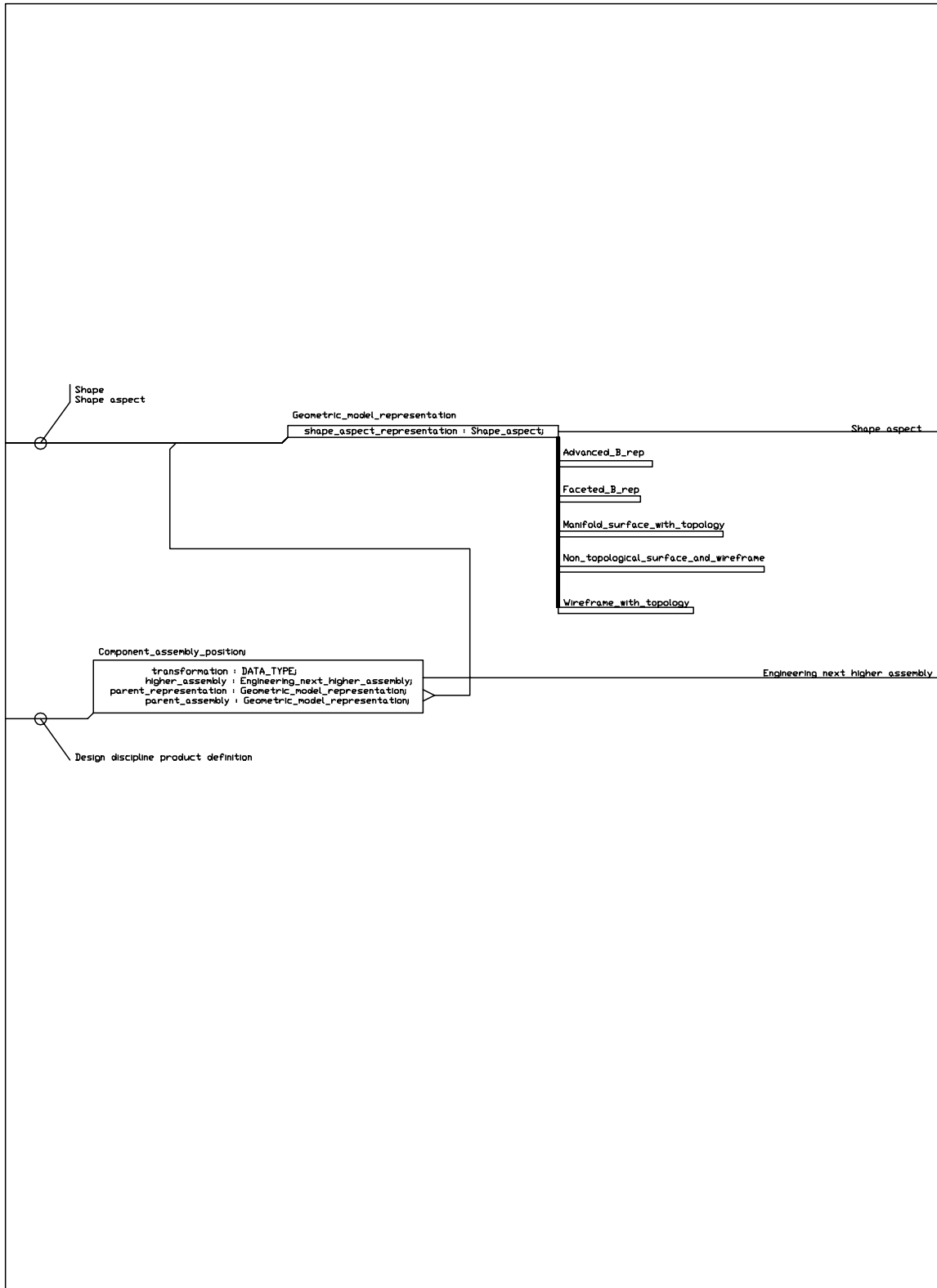


Figure 4 – Shape

:

2.3.3 Advanced_B_rep

An **Advanced_B_rep** is a type of **Geometric_model_representation** (see 2.3.1) that represents the shape of a part or an aspect of the shape of a part by an advanced-boundary-representation solid model. This representation allows for the definition of curves and surfaces and for the topology that bounds them. Boundaries are explicitly defined only by topology. All of the geometry that defines the part's shape or an aspect of the part's shape shall be associated with topology.

EXPRESS specification:

```
*)  
ENTITY Advanced_B_rep  
SUBSET OF (Geometric_model_representation);  
END_ENTITY;  
(*
```

2.3.4 Faceted_B_rep

A **Faceted_B_rep** is a type of **Geometric_model_representation** (see 2.3.1) that represents the shape of a part or an aspect of the shape of a part by a faceted boundary representation solid model. This representation allows for the definition of shapes represented by planar surfaces as the bounding surfaces. Only points and planar polygons are used in this representation. Much of the topology information is implicit for this representation. Shells consist of faces bounded exclusively by polygons.

EXPRESS specification:

```
*)  
ENTITY Faceted_B_rep  
SUBTYPE OF (Geometric_model_representation);  
END_ENTITY;  
(*
```

2.3.5 Manifold_surface_with_topology

A **Manifold_surface_with_topology** is a type of **Geometric_model_representation** (see 2.3.1) that represents the shape of a part or an aspect of the shape of a part using manifold surfaces with topology. 3D curves, surfaces, and topology are used to define the outer boundary of the part.

EXPRESS specification:

```
*)  
ENTITY Manifold_surface_with_topology  
SUBTYPE OF Geometric_model_representation;  
END_ENTITY;  
(*
```

2.3.6 Non_topological_surface_and_wireframe

A **Non_topological_surface_and_wireframe** is a type of **Geometric_model_representation** (see 2.3.1) that represents the shape or portions of the shape of a part using surface or wireframe geometry without topology. These representations are formed by the use of points, curves and surfaces only. The boundaries of the curves are defined explicitly by points on the curves and explicit associations between the points and the curves that they bound. The boundaries of the surfaces are defined by curves on the surfaces and explicit associations between the curves and the surfaces that they bound. Surfaces and curves must be explicitly trimmed unless they are closed.

EXPRESS specification:

```
*)
ENTITY Non_topological_surface_and_wireframe
SUBTYPE OF (Geometric_model_representation);
END_ENTITY;
(*
```

2.3.7 Wireframe_with_topology

A **Wireframe_with_topology** is a type of **Geometric_model_representation** (see 2.3.1) that represents the shape or a portion of the shape of a part using wireframes that define an implicit volume or are trimmed by edge topology. This includes 3D curves and topology that define a graph of vertices and edges.

EXPRESS specification:

```
*)
ENTITY Wireframe_with_topology
SUBTYPE OF (Geometric_model_representation);
END_ENTITY;
(*
```

2.4 Effectivity and End-Item Identification

The Effectivity and End-Item Identification UoF contains information about the planned usage of components in a product model. It also contains information about the different products that are sold by an organization to its customers. It provides the information structures for an organization to control the configuration of these products.

The graphic representation of the Effectivity and End-Item Identification UoF is shown in Figure 5.

2.4.1 Planned_effectivity

A **Planned_effectivity** is the intended use of a part in a particular configuration of a product.

EXPRESS specification:

```
*)
ENTITY Planned_effectivity
```

:

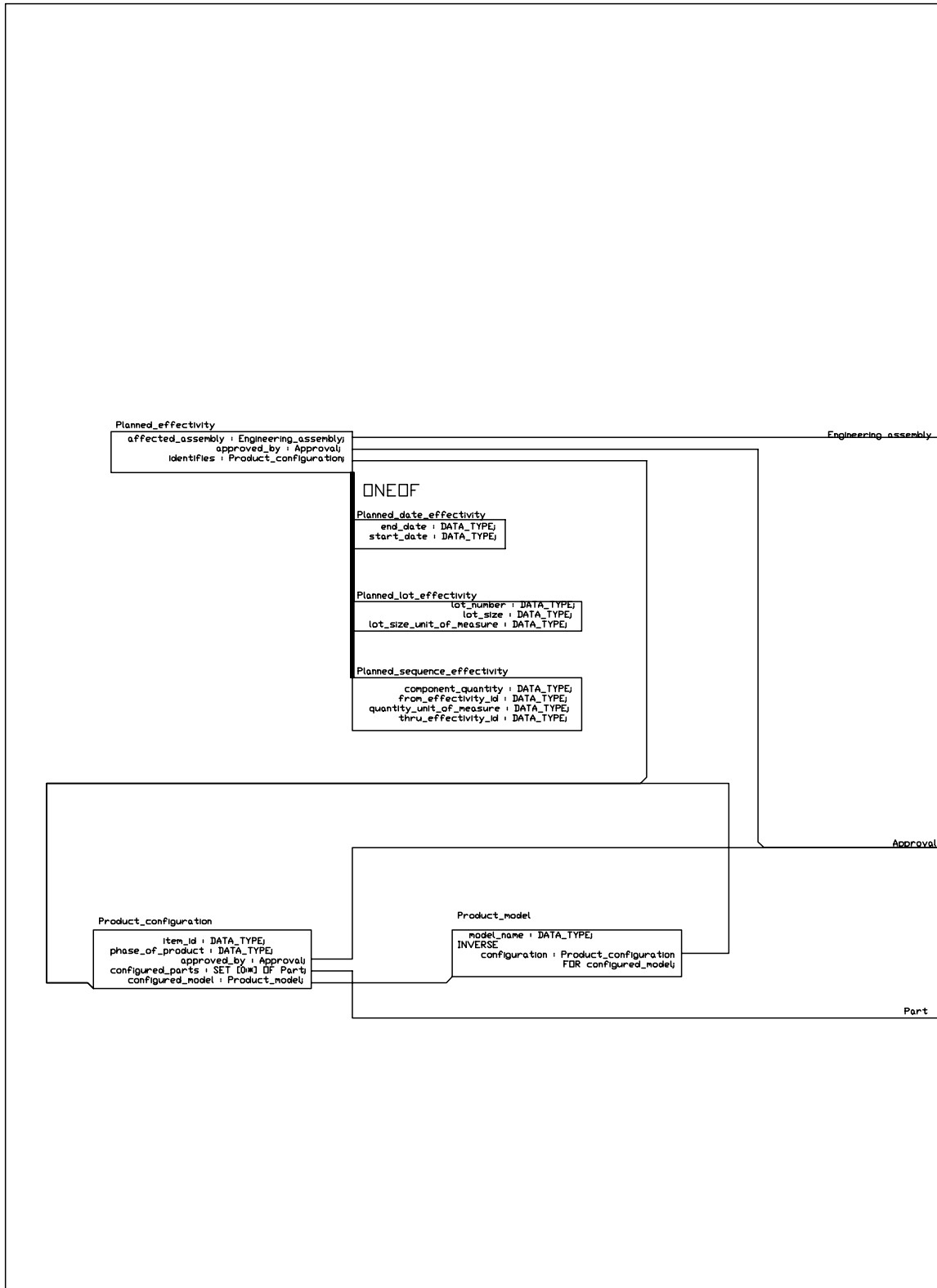


Figure 5 – Effectivity and End-Item Identification

:

```
SUPERTYPE OF ONEOF (Planned_date_effectivity,  
                    Planned_lot_effectivity,  
                    Planned_sequence_effectivity);  
    affected_assembly : Engineering_assembly;  
    approved_by : Approval;  
    identifies : Product_configuration;  
END_ENTITY;  
(*
```

Attribute definitions:

affected_assembly: the **Engineering_assembly** in which this **Planned_effectivity** is effective.

approved_by: the **Approval** which approves this **Planned_effectivity**.

identifies: the **Product_configuration** which this **Planned_effectivity** identifies.

2.4.2 Planned_date_effectivity

A **Planned_date_effectivity** is a type of **Planned_effectivity** (see 2.4.1). It is the specification by the design organization of the expected usage of a part in a **Product_configuration**. The expected usage of the part within the product configuration is determined by one or two associated dates.

EXPRESS specification:

```
*)  
ENTITY Planned_date_effectivity  
SUBTYPE OF Planned_effectivity);  
    end_date : DATA_TYPE;  
    start_date : DATA_TYPE;  
END_ENTITY;  
(*
```

Attribute definitions:

end_date: the last date on which the part is planned by a design organization to be used in the **Product_configuration**. The **end_date** need not be specified for a particular **Planned_date_effectivity**. No **end_date** at the time of exchange signifies that an **end_date** has not yet been identified.

start_date: the first date on which the part is to be used in the **Product_configuration**.

2.4.3 Planned_lot_effectivity

A **Planned_lot_effectivity** is a type of **Planned_effectivity** (see 2.4.1). It is the specification of the use of a part in a **Product_configuration** where the part is produced as one of a group. Lots are used when parts are produced in batches or when important characteristics might vary between production runs.

EXPRESS specification:

:

```
*)
ENTITY Planned_lot_effectivity
SUBTYPE OF (Planned_effectivity);
    lot_number : DATA_TYPE;
    lot_size : DATA_TYPE;
    lot_size_unit_of_measure : DATA_TYPE;
END_ENTITY;
(*
```

Attribute definitions:

lot_number: the identification of the group of parts that compose a lot.

lot_size: the quantity of parts within the lot.

lot_size_unit_of_measure: the fixed quantity amount in terms of which the **lot_size** is expressed.

2.4.4 Planned_sequence_effectivity

A **Planned_sequence_effectivity** is a type of **Planned_effectivity** (see 2.4.1). It is the specification of the intended use by a design organization of a part within a range of **Product-configurations** identified by planned serial numbers.

EXPRESS specification:

```
*)
ENTITY Planned_sequence_effectivity
SUBTYPE OF (Planned_effectivity);
    component_quantity : DATA_TYPE;
    from_effectivity_id : DATA_TYPE;
    quantity_unit_of_measure : DATA_TYPE;
    thru_effectivity_id : OPTIONAL DATA_TYPE;
END_ENTITY;
(*
```

Attribute definitions:

component_quantity: the number of units that are effective for a particular part in a configuration.

from_effectivity_id: the beginning planned serial number of the range for the **Planned_sequence_effectivity**.

quantity_unit_of_measure: the measure in terms of which the **component_quantity** is expressed.

EXAMPLES

10 – **Quantity_unit_of_measure** may be rolls, sheets, or bars, among other things.

:

thru_effectivity_id: the last planned serial number of the range for the **Planned_sequence_effectivity**. The **thru_effectivity_id** need not be specified for a particular **Planned_sequence_effectivity**.

2.4.5 Product_configuration

A **Product_configuration** is a variation of a **Product_model** (see 2.4.6). Configuration management is based on this application object.

EXAMPLES

11 – A **Product_configuration** for the F14 military aircraft is the D configuration of the F14. The organization has defined four configurations of the F14, A,B,C and D. D, in this instance, is the configuration. A single product model might have many different configurations in order for the organization to identify different variations of it.

EXPRESS specification:

```
*)
ENTITY Product_configuration
    item_id : DATA_TYPE;
    phase_of_product : DATA_TYPE;
    approved_by : Approval;
    configured_parts : SET [0:*) OF Part;
    configured_model : Product_model;
END_ENTITY;
(*
```

Attribute definitions:

item_id: the unique identification of a variation of the a **Product_model**.

phase_of_product: the stage in the life cycle of the product in which a particular version of the design is being exchanged.

EXAMPLES

12 – Within an organization four phases of a product may be Technology and Concept Development, Development and Refinement, Manufacturing and Assembly Validation, and Production.

configured_model: the **Product_model** which this **Product_configuration** configures.

2.4.6 Product_model

A **Product_model** is the product that the organization provides to its customers. The **Product_model** is identified for planning purposes in the design stage of a product.

EXPRESS specification:

```
*)
ENTITY Product_model
    model_name : DATA_TYPE;
```

:

```
INVERSE
  configuration : Product_configuration
                FOR configured_model;
END_ENTITY;
(*
```

Attribute definitions:

model_name: the unique identification assigned by an organization to a product that the organization provides to its customers.

EXAMPLES

13 – F14 is the identification of a **Product_model** that is a military aircraft.

configuration: the **Product_configuration** which configures this **Product_model**.

2.5 Design Activity Control

The Design Activity Control UoF contains the information which describes the history of part versions. It identifies initial part design requirements as well as the change requirements for revised part versions. It provides the directive for work to proceed in the development of these initial or modified part versions.

The graphic representation of the Design Activity Control UoF is shown in Figure 6.

2.5.1 Change_order

A **Change_order** is a type of **Work_order** (see 2.5.5) that authorizes the development of a modified design of a part that results in the establishment of a new part version.

EXPRESS specification:

```
*)
ENTITY Change_order
SUBTYPE OF (Work_order);
  adopted_solution : DATA_TYPE;
  change_date      : DATA_TYPE;
END_ENTITY;
(*
```

Attribute definitions:

adopted_solution: the solution selected from the set of recommended solutions for a **Change-request** (see 2.5.2).

change_date: the date when the **Change_order** became effective.

2.5.2 Change_request

A **Change_request** is a type of **Work_request** (see 2.5.6) that identifies the proposed extent of work to be done to modify the design of a part.

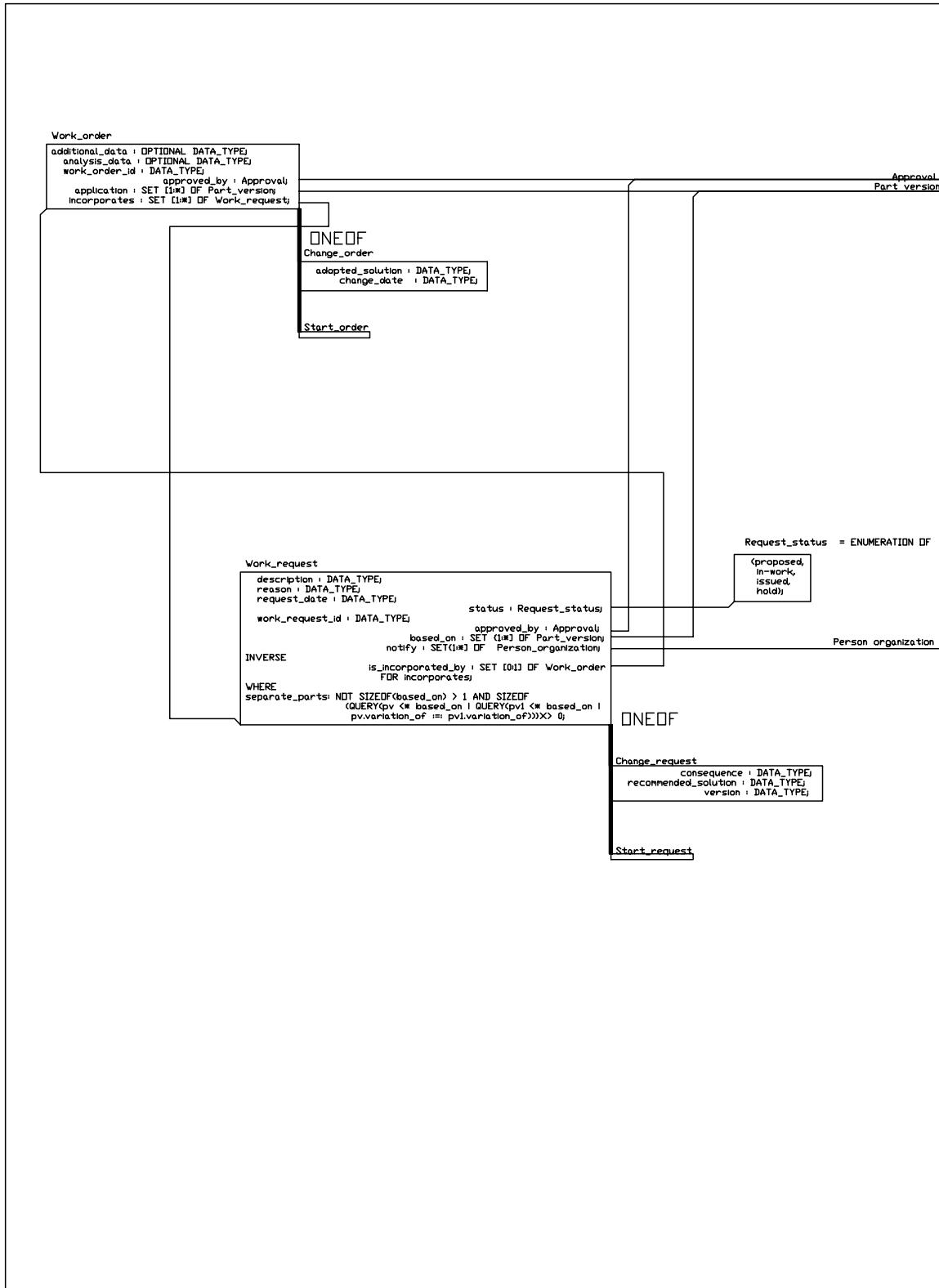


Figure 6 – Design Activity Control

:

EXPRESS specification:

```
*)  
ENTITY Change_request  
SUBTYPE OF Work_request;  
    consequence : DATA_TYPE;  
    recommended_solution : DATA_TYPE;  
    version : DATA_TYPE;  
END_ENTITY;  
(*
```

Attribute definitions:

consequence: the effects in performance, functionality, or form of a particular recommended solution on the affected part version. There may be more than one consequence for a **Change-request**.

EXAMPLES

14 – A change to a design might be to thicken a support on the wing of an aircraft. The consequence of the additional thickness would be that the wing will now support x amount of additional load and the aircraft would be carrying y amount of additional weight.

recommended_solution: a potential resolution that would satisfy the requirements described in the change request through the **Work_request** description (see 2.5.6). There may be more than one **recommended_solution** for a **Change_request**.

version: a unique identifier for each iteration of a change request.

2.5.3 Start_order

A **Start_order** is a type of **Work_order** (see 2.5.5) that authorizes work in the development of the initial design of a part, resulting in the creation of an original part version.

EXPRESS specification:

```
*)  
ENTITY Start_order  
SUBTYPE OF (Work_order);  
END_ENTITY;  
(*
```

2.5.4 Start_request

A **Start_request** is a type of **Work_request** (see 2.5.6) that identifies the work to be done to initiate the design of a part.

EXPRESS specification:

```
*)  
ENTITY Start_request  
SUBTYPE OF (Work_request);  
END_ENTITY;  
(*
```

2.5.5 Work_order

A **Work_order** is a document that authorizes work in the development of an initial or modified design of a part. A **Work_order** is the result of the processing of one or many **Work_requests** (see 2.5.6).

EXPRESS specification:

```
*)
ENTITY Work_order
  SUPERTYPE OF ONEOF(Start_order,
                     Change_order;
additional_data : OPTIONAL DATA_TYPE;
  analysis_data : OPTIONAL DATA_TYPE;
  work_order_id : DATA_TYPE;
  approved_by : Approval;
  application : SET [1:~] OF Part_version;
  incorporates : SET [1:~] OF Work_request;
END_ENTITY;
(*
```

Attribute definitions:

additional_data: additional pertinent information that was compiled during the development of the related set of proposed work requests. The **additional_data** need not be specified for a particular **Work_order**.

analysis_data: the results of an assessment that was performed to validate the design or modification in a related set of proposed work requests. The **analysis_data** need not be specified for a particular **Work_order**.

work_order_id: a unique identifier for the work authorized by the **Work_order**.

approved_by: the **Approval** which approves this **Work_order**.

application: the set of **Part_versions** to which this **Work_order** applies.

incorporates: the set of **Work_requests** which this **Work_order** incorporates.

2.5.6 Work_request

A **Work_request** is a document that identifies the work to be done to initiate or redefine the design of a part.

EXPRESS specification:

```
*)
ENTITY Work_request
  SUPERTYPE OF ONEOF (Start_request,
                     Change_request);
  description : DATA_TYPE;
  reason : DATA_TYPE;
  request_date : DATA_TYPE;
```

:

```
status : Request_status;
work_request_id : DATA_TYPE;
approved_by : Approval;
based_on : SET {1:*} OF Part_version;
notify : SET{1:*} OF Person_organization;
INVERSE
  is_incorporated_by : SET [0:1] OF Work_order
                      FOR incorporates;
WHERE
separate_parts: NOT SIZEOF(based_on) > 1 AND SIZEOF
  (QUERY(pv <* based_on | QUERY(pv1 <* based_on |
    pv.variation_of :=: pv1.variation_of)))> 0;
END_ENTITY;
(*
```

Attribute definitions:

description: an explanation of the work to be done to develop or redefine the design of a part.

reason: the purpose for generating the **Work_request**.

request_date: the date when the **Work_request** was created.

status: the current level of completion of the **Work_request**.

work_request_id: a unique identifier for the proposed work on a part's design.

approved_by: the **Approval** object which approves this **Work_request**.

based_on: the set of **Part_versions** on which this **Work_request** is based.

notify: the set of **Person_organizations** which receive this **Work_request**.

is_incorporated_by: the **Work_order** which may incorporate this **Work_request**.

Formal propositions:

separate_parts: If a **Work_request** requests work for greater than one **Part_version**, these **Part_version** objects shall be of separate **Part** objects.

2.5.7 Request_status

This enumeration type specifies valid values for **Work_request status**.

EXPRESS specification:

```
*)
TYPE Request_status = ENUMERATION OF
  (proposed,
   in-work,
   issued,
   hold};
```

:

END_TYPE;
(*

2.6 Bill of Material

The Bill of Material UoF contains the objects necessary to represent the structured list of materials or components required to build a part.

The graphic representation of the Bill of Material UoF is shown in Figure 7.

2.6.1 Engineering_assembly

An **Engineering_assembly** is the parent-child relationship between an assembly and a component or sub-assembly.

NOTE – The scope of this part of ISO 10303 is limited to the exchange of product design definition bill of materials. It does not cover all bills of material required by an organization.

EXPRESS specification:

```
*)  
ENTITY Engineering_assembly  
SUPERTYPE OF (Engineering_next_higher_assembly,  
               Engineering_promissory_usage;  
  security_code : DATA_TYPE;  
    defined_by : Design_discipline_product_definition;  
  substitutes : SET [0:*) OF Substitute_part;  
END_ENTITY;  
(*
```

Attribute definitions:

security_code: the security classification of a component in an assembly.

NOTE – This differs from the attribute **security_code** associated with **Part_version** (see 2.1.4) in that some aspect of the component as it relates to an assembly changes its security classification when that part exists as the component within the assembly to which this security code is related. The part may have a security code when it exists alone that is independent of its use in any particular assembly. The security code specified here, however, defines the security code of the part in this very specific context.

EXAMPLES

15 – A wheel may be used in the production of a number of automobiles and its security classification is unclassified. That same wheel is used in an assembly that is then used in an experimental, highly sensitive, new automobile that is under development. Although the wheel itself is unclassified, the assembly that is in the new automobile is classified as top secret and the fact that this particular wheel is in the assembly also needs to be classified as top secret. This attribute then redefines the security classification of the wheel to be top secret when it is used as a component in the assembly for this particular automobile under development.

defined_by: the **Design_discipline_product_definition** which defines this **Engineering_assembly**.

:

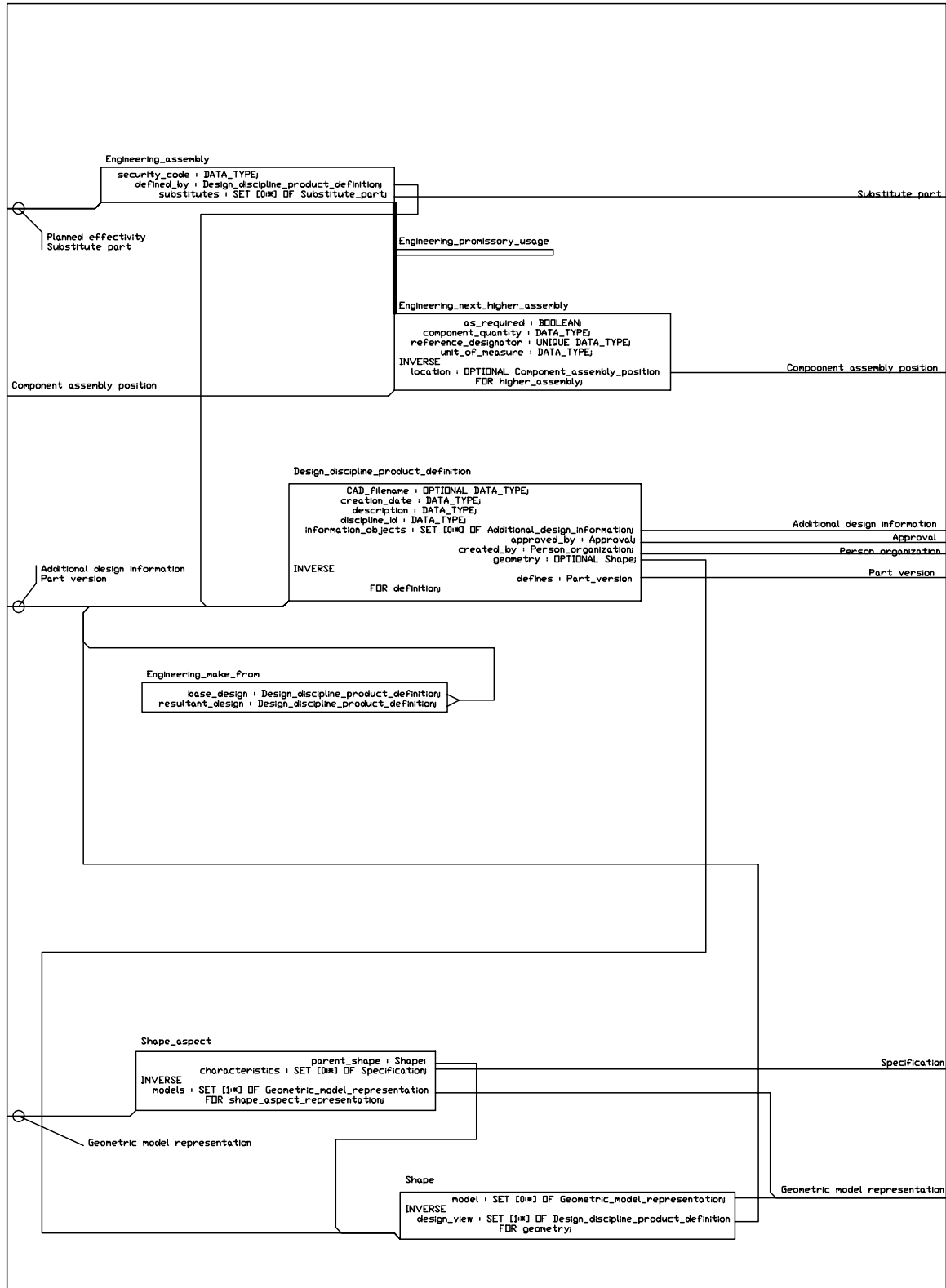


Figure 7 – Bill of Material

:

substitutes: the set of **Substitute_parts** which substitute for **Parts** in this Engineering-assembly.

2.6.2 Design_discipline_product_definition

A **Design_discipline_product_definition** is one of the organizational definitions or views of a **Part_version** (see 2.1.4).

NOTE – This entity may be used to capture the definition of a particular **Part_version** at any intermediate stage of its development where the definition is not yet formally tracked by an organization at the **Part_version** level. It may be used to capture the various stages in the definition cycle of a product.

EXAMPLES

16 – Stages in the definition cycle may be a bill of material view, a finite element model view, a functional system view, or a manufacturing view.

EXPRESS specification:

*)

```
ENTITY Design_discipline_product_definition
    CAD_filename : OPTIONAL DATA_TYPE;
    creation_date : DATA_TYPE;
    description : DATA_TYPE;
    discipline_id : DATA_TYPE;
    information_objects : SET [0:*) OF Additional_design_information;
    approved_by : Approval;
    created_by : Person_organization;
    geometry : OPTIONAL Shape;
```

INVERSE

```
    defines : Part_version
        FOR definition;
```

END_ENTITY;

(*

Attribute definitions:

CAD_filename: the name of the file that contains the geometric part description in a computer aided design system. The **CAD_filename** need not be specified for a particular **Design_discipline_product_definition**; if this name is provided, the file is assumed to contain an external definition of the shape of the part that is meaningful only within the organization that designed the part.

creation_date: the date and time that the **Design_discipline_product_definition** was first defined.

description: the purpose for a particular definition of the product.

discipline_id: the view or stage to which the definition of the product pertains.

information_objects: the set of **Additional_design_information** which apply to this **Design_discipline_product_definition**.

:

approved_by: the **Approval** which approves this **Design_discipline_product_definition**.

created_by: the **Person_organization** which created this **Design_discipline_product_definition**.

geometry: the **Shape** which may define the geometric characteristics of this **Design_discipline_product_definition**.

defines: the **Part_version** which this **Design_discipline_product_definition** defines.

2.6.3 Engineering_make_from

An **Engineering_make_from** is the relationship between two parts in which one part is used as the basis for the design of the other part. This relationship is established by design so that it may be tracked through the life cycle of the product.

EXAMPLES

17 – Company A designs an assembly using a cup holder from company B but adds two additional mounting holes to company B's design. In this case, company A identifies the cup holder designed by company B as an **Engineering_make_from** part.

EXPRESS specification:

*)

```
ENTITY Engineering_make_from
    base_design : Design_discipline_product_definition;
    resultant_design : Design_discipline_product_definition;
END_ENTITY;
(*
```

Attribute definitions:

base_design: the **Design_discipline_product_definition** which is the base design of this **Engineering_make_from**.

resultant_design: the **Design_discipline_product_definition** which is the resultant design of this **Engineering_make_from**.

2.6.4 Engineering_next_higher_assembly

An **Engineering_next_higher_assembly** is a type of **Engineering_assembly** (see 2.6.1). It is the relationship of a part to its immediate parent within an assembly hierarchy.

EXPRESS specification:

*)

```
ENTITY Engineering_next_higher_assembly
SUBTYPE OF (Engineering_assembly);
    as_required : BOOLEAN;
    component_quantity : DATA_TYPE;
    reference_designator : UNIQUE DATA_TYPE;
```

:

```
        unit_of_measure : DATA_TYPE;
INVERSE
    location : OPTIONAL Component_assembly_position
        FOR higher_assembly;
END_ENTITY;
(*
```

Attribute definitions:

as_required: an attribute specifying whether the component quantity is explicitly defined (TRUE) or is to be quantified as needed in a particular usage (FALSE).

EXAMPLES

18 – If the amount of sheet metal that is to be consumed in making a part is to be identified, one roll might be specified, where one is the quantity and roll is the unit of measure. Rolls of sheet metal might also be used as required. In this case, as required is the quantity and roll is again the unit of measure.

component_quantity: the count of a component part found in an immediately higher assembly if **as_required** is false.

reference_designator: a unique identifier that distinguishes a particular instance of a component in an assembly where more than one of a particular component part is used in an assembly.

EXAMPLES

19 – If four identical wheels are used in a car assembly then the **reference_designator** distinguishes the left-front wheel, for example, from the others.

unit_of_measure: the fixed quantity in terms of which the **component_quantity** is expressed.

location: the **Component_assembly_position** which is the location of this **Engineering-next_higher_assembly**.

2.6.5 Engineering_promissory_usage

An **Engineering_promissory_usage** is a type of **Engineering_assembly** (see 2.6.1). It is the relationship of a part to a higher assembly where the next higher assembly of the part has not been defined.

EXAMPLES

20 – A number of brackets are needed in order to support a beam structure in the fuselage of an aircraft. A designer is designing one of the brackets and needs to specify its intended use in the product structure. The designer does not yet know the next higher assembly since the beam to which the bracket is to be attached is not yet defined in the product structure. The designer does, however, know the fuselage in which the bracket is used. Using this object, the designer can specify that the bracket is promised within a sub-assembly in support of the fuselage.

:

EXPRESS specification:

```
*)
ENTITY Engineering_promissory_usage
SUBTYPE OF (Engineering_assembly);
END_ENTITY;
(*
```

2.6.6 Shape

A **Shape** is the mathematical representation of the form of a part.

EXPRESS specification:

```
*)
ENTITY Shape
    model : SET [0:~] OF Geometric_model_representation;
INVERSE
    design_view : SET [1:~] OF Design_discipline_product_definition
                    FOR geometry;
END_ENTITY;
(*
```

Attribute definitions:

design_view: the **Design_discipline_product_definitions** which have their geometric characteristics defined by this **Shape**.

model: the **Geometric_model_representation** which is the representation of this **Shape**.

2.6.7 Shape_aspect

A **Shape_aspect** is a distinct portion of the shape of a part. The shape of a part consists of one or many **Shape_aspects**.

NOTE – Often in design a number of different types of geometric models will define the shape of a single part depending on the complexity of the shape of the part in different physical areas. Each of these physical areas may be identified as a **Shape_aspect**.

EXPRESS specification:

```
*)
ENTITY Shape_aspect
    parent_shape : Shape;
    characteristics : SET [0:~] OF Specification;
INVERSE
    models : SET [1:~] OF Geometric_model_representation
                FOR shape_aspect_representation;
END_ENTITY;
(*
```

Attribute definitions:

parent_shape: the **Shape** of which this **Shape_aspect** is a portion.

:

characteristics: the set of **Specifications** which specify a characteristics of this **Shape_aspect**.

models: the set of **Geometric_model_representations** which represent this **Shape_aspect**.

2.7 Authorization

The Authorization UoF contains the constructs to indicate the acceptance of a piece of product data, on a specific date, at a particular level of status, by a group of people having roles that allow them to grant the authorization.

The graphic representation of the Authorization UoF is shown in Figure 8.

2.7.1 Approval

An **Approval** is an indication within an organization of concurrence or nonconcurrence with a piece of product data.

EXPRESS specification:

```
*)  
ENTITY Approval;  
    date : DATA_TYPE;  
    purpose : DATA_TYPE;  
    status : DATA_TYPE;  
    authorized_by : SET [1:*) OF Person_organization;  
END_ENTITY;  
(*
```

Attribute definitions:

date: the approval became, or will become, effective.

purpose: the reason for the approval.

status: the state of consent applied to a piece of product data or a relationship between pieces of product data. Valid values for the status of an approval are: approved and not approved. Approved means the required concurrence has been established. Not approved means the required concurrence has not been established.

authorized_by: the set of **Person_organizations** authorizing this **Approval**.

2.7.2 Person_organization

A **Person_organization** is the identification of an individual and the particular organization within which the individual plays a role.

EXPRESS specification:

```
*)  
ENTITY Person_organization
```

:

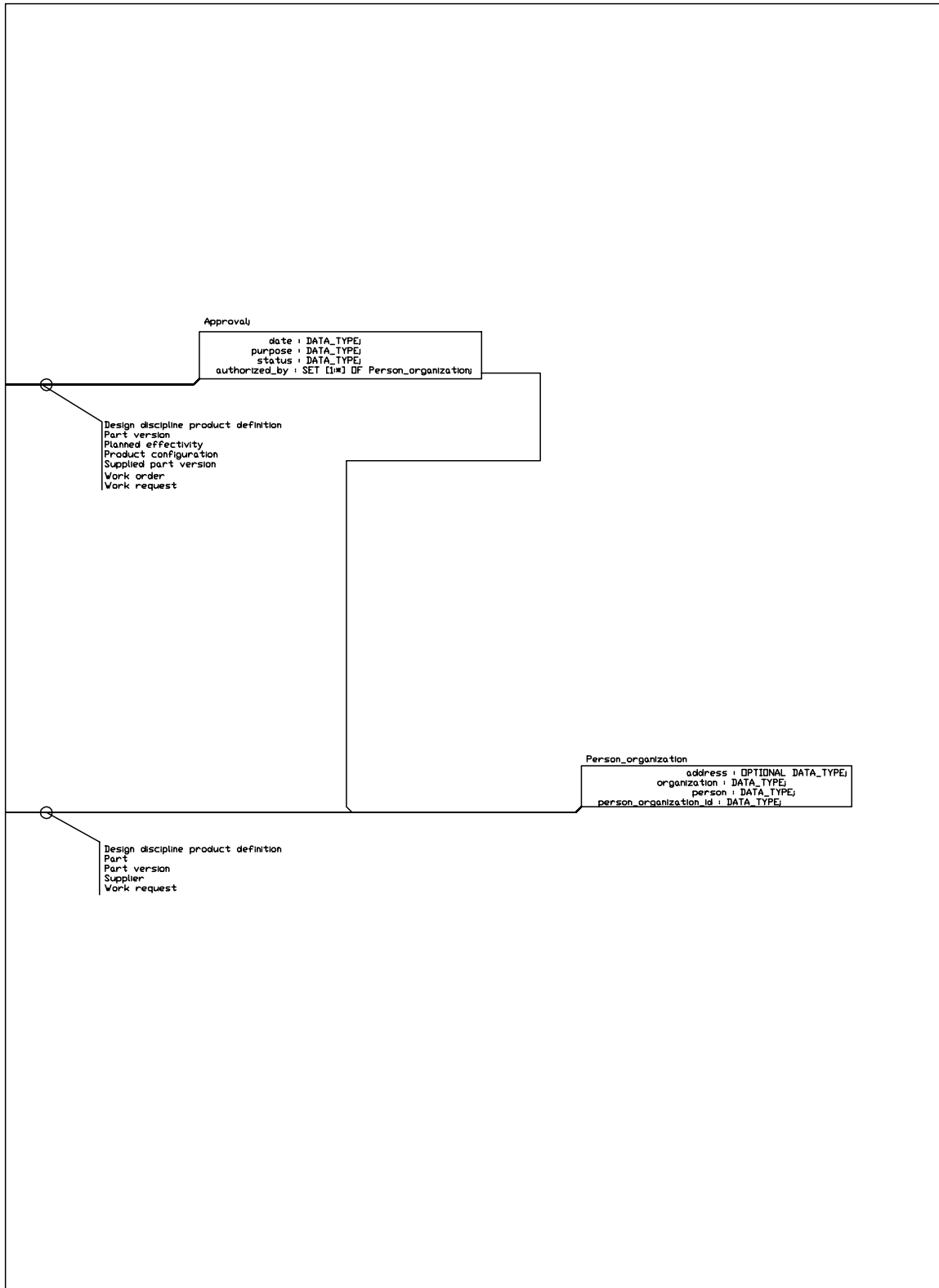


Figure 8 – Authorization

:

```
        address : OPTIONAL DATA_TYPE;
    organization : DATA_TYPE;
        person : DATA_TYPE;
    person_organization_id : DATA_TYPE;

END_ENTITY;

END_SCHEMA;
(*)
```

Attribute definitions:

address: the routing for paper and electronic mail and, in some instances, the physical location of a person and organization. The address need not be specified for a particular **Person-organization**.

organization: a collection of people grouped together for one or more common purposes.

person: an individual human being.

person_organization_id: the unique identification of the **Person-organization**.

3 The 203 ARM Objects

The following table shows each ARM object, its location, and those objects which either reference it or are subtypes of it.

:

OBJECT NAME	LOCATION	REFERENCE	SUBTYPE
Additional design information	2	Design discipline product definition	
Advanced B rep	3		
Alternate part	1		
Approval	7	Design discipline product definition Part version Planned effectivity Product configuration Supplied part version Work order Work request	
Change order	5		
Change request	5		
Component assembly position	3		
Design discipline product definition	6	Additional design information Part version Shape	
Design specification	2		
Engineering assembly	6	Planned effectivity Substitute part	Engineering next higher assembly Engineering promissory usage
Engineering make from	6		
Engineering next higher assembly	6	Component assembly position	
Engineering promissory usage	6		
Faceted B rep	3		
Geometric model representation	3	Component assembly position Shape Shape aspect	Advanced B rep Faceted B rep Manifold surface with topology Non topological surface and wireframe Wireframe with topology
Make or buy	1	Part version	
Manifold surface with topology	3		
Material specification	2		
Non topological surface and wireframe	3		

:

OBJECT NAME	LOCATION	REFERENCE	SUBTYPE
Part	1	Alternate part Part version Product configuration Substitute part	Substitute part
Part type enum	1	Part	
Part version	1	Design discipline product definition Part Supplied part version Work order Work request	
Person organization	7	Approval Design discipline product definition Part Part version Supplier Work request	
Planned date effectivity	4		
Planned effectivity	4		Planned date effectivity Planned lot effectivity Planned sequence effectivity
Planned lot effectivity	4		
Planned sequence effectivity	4		
Process specification	2		
Product configuration	4	Planned effectivity Product model	
Product model	4	Product configuration	
Request status	5	Work request	
Shape	6	Design discipline product definition Shape aspect	
Shape aspect	6	Geometric model representation	
Specification	2	Additional design information Shape aspect Usage constraint	Design specification Material specification Process specification Surface finish specification
Start order	5		
Start request	5		
Status of release	1	Part version	
Substitute part	1	Engineering assembly	

:

OBJECT NAME	LOCATION	REFERENCE	SUBTYPE
Supplied part version	1	Part version	
Supplier	1	Supplied part version	
Surface finish specification	2		
Usage constraint	2	Specification	
Wireframe with topology	3		
Work order	5	Work request	Change order Start order
Work request	5	Work order	Change request Start request